非タスク指向型対話システムでの話題同定のためのコーパス構築に関する研究

著者	柴田 尚樹
内容記述	筑波大学修士(情報学)学位論文・平成31年3月25日授与(41296号)
発行年	2019
URL	http://hdl.handle.net/2241/00159777

非タスク指向型対話システムでの話題同定のためのコーパス構築に関する研究

筑波大学
図書館情報メディア研究科
2019年3月
柴田　尚樹

目 次

表 目 次

図 目 次

第1章　はじめに

近年，スマートスピーカーやスマートフォン，タブレット型端末の普及に伴い，音声アシスタント，雑談対話システムなど，様々な種類の対話システムが開発されている．これら対話システムは，タスクの達成を目的とする対話システム（タスク指向型対話システム）と雑談を行うための対話システム（非タスク指向型対話システム）とに区別される．また，両者の機能を備えた対話システムも存在しており，例えば，スマートスピーカーにおいては，ユーザの指示に従い連動した部屋の照明を消す機能，音楽を流す機能といったタスク指向型対話システムの機能に加えて，雑談を行う非タスク指向型対話システムの機能を備えるものが存在する．

ユーザの日常に対話システムがより深く溶け込んでいくためには，非タスク指向型対話システムの発展が重要である．小磯ら [1] は 1 日の会話行動の種類と従事時間について調査を行い，いずれの年代，性別，職業においても，雑談が 50%以上を占めると報告している．このことから，対話システムに対する人間の信頼感を向上させ，自然なヒューマンマシンインタフェースとして認識されるためには，対話システムが雑談に対応する機能を備えることが重要な課題だといえる．

ユーザと自然なやり取りを行う非タスク指向型対話システムを実現する上では，幅広い話題への対応が課題となる．ユーザがある話題について話しかけた際に，非タスク指向型対話システムがその話題を扱うことができない場合，対話システムの応答は的を射ないものとなる．この対策として，個人によるルールベースでの話題への対応が挙げられるが，話題となる言葉は日々増加するため，それらに個人で対応することは困難である．

そこで，本研究では Wikipedia[1] に着目する．Wikipedia は，幅広い話題の記事を扱い，利用者によって日々記事が更新されていく性質を持つ．このため，Wikipedia を利用することで多くの話題を扱う非タスク指向型対話システムの実現が期待できる．特に，本研究では，Wikipedia 記事から有益な記述を抽出して発話に利用するというアプローチを考える．有益な記述は，対話システムが会話内容を踏まえて次の発話を生成する際に利用できるためである．このような手法を実現するためには人間が何を有益と判断するかについて分析する必要がある．しかし，現状ではこのような分析の対象として用いることができるコーパスが整備されておらず，このことが，Wikipedia を非タスク指向型対話システムの話題として利用する研究を困難としている．

[1] https://ja.wikipedia.org

このことを踏まえて，本研究では，会話中の発話を “会話している人らにとって有益な記述” が存在する Wikipedia 記事に結びつけるコーパスの構築に取り組む．本研究では，Wikipedia 記事中の任意の長さの文字列を “記述” と呼ぶ．コーパスの仕様については 3.1 節で述べるが，発話単位で話題を表す Wikipedia 記事に結びつけるため，このコーパスは会話における話題の同定や，話題の変化を追う機械学習に利用できる．また，このコーパスは，1 つの発話に 1 つも記事を結び付けない場合や，複数の記事を結びつける場合も想定したコーパスとする．1 つの発話に 1 つも記事を結び付けない発話を想定することで，挨拶のみの会話など，Wikipedia を参照する必要がない会話を判別する機械学習に利用できる．1 つの発話に複数の話題を結びつけることを想定することで，発話生成時に，複数の話題候補を考慮した対話システムの構築が期待できる．Wikipedia 記事が有益である場合には，有益な記述についてもコーパスに収録することとする．これにより，本研究で構築するコーパスは，記述単位での話題同定を行う機械学習での利用が可能となり，この機械学習によって同定される有益な記述は，非タスク指向型対話システムの発話生成時に利用できるものとなる．

また，コーパス構築において，会話データの収集や専門家によるアノテーションにはコストがかかる．本研究ではあらかじめ収集された Twitter の会話データに対して，クラウドソーシングによるアノテーションを行うことで，客観性を担保しつつコストを抑えたコーパスの構築を試みる．Twitter 会話データは，コーパス構築に際し集められる会話データとは異なり，自然発生した会話からなる会話データである．そのため，Twitter 会話データには，例えば挨拶のみの会話など，日常的な会話ではあるが Wikipedia に結びつける必要がない会話が含まれていることからも，このような会話にまでアノテーションを行ったデータは，日常的な対話をこなす対話システムを検討する際に有用であると考えられる．

アノテーションに関しては，アノテーター 1 人の主観のみに基づくアノテーションでは，客観性に欠けたものとなる危惧がある．加えて，何が有益であるかの判断は人により異なると考えられるため，ある人が有益であるとする記述を，他の人は有益ではないとすることがありうる．そのため，本研究ではクラウドソーシングを 2 段階に分けて適用し，クラウドソーシングのワーカーが Wikipedia 記事が有益であるか判断をし，有益であると判断する場合には有益な記述の抽出も行う段階と，再びクラウドソーシングを利用し，前段階でワーカーが抽出した有益な記述についてどれだけの合意が得られるかを確認する段階とを設ける．そして，有益な記述に対する合意の度合いについてもコーパスに保存することで，記述単位での話題同定を行う機械学習を利用した非タスク指向型対話システムにおいて，有益な記述に対する合意の度合いに基づく発話生成ができると考えられる．

本論文の構成は以下の通りである．第 2 章では，非タスク指向型対話システムや Wikipedia について関連研究を挙げる．第 3 章では，本研究でのコーパス構築方法について述べる．第 4 章では、本研究で作成するコーパスを分析し，考察を述べる．第 5 章では，結論を述べる．

第2章　関連研究

2.1　Wikipediaへの対応付けに関する研究

本研究で構築する非タスク指向型対話システムでの利用に向けた話題同定のためのコーパスは，エンティティ・リンキング（Entity Linking）という枠組みと関連しており，そのうち特に，Wikification [2] と関連する．Entity Linking とは，文章からキーワード（メンション）を抽出し，メンションを知識ベースの対応するエンティティに結びつけることである．Entity Linking は，知識ベースに Wikipedia を利用した場合，Wikification と呼ばれる．本研究は，Twitter における会話中の話題を Wikipedia の記事に結びつけ同定する試みであることから，Wikification と関連する．

日本語における Wikification は，日本語 Wikification として研究されている．Jargalsaikhan ら [3] は，拡張固有表現に Wikipedia 記事へのリンクを付与したコーパスの開発に取り組んでいる．松田ら [4] は日本語 Wikification コーパス [3] を訓練データに利用した学習による，日本語 Wifikicaion のためのツールを開発している．Murawaki ら [5] もまた，日本語における Wikification コーパスの構築に取り組んでいる．こちらは，固有表現に限らず一般的なフレーズも Wikification の対象としている．

Wikification の発展は，様々な自然言語処理分野の発展に寄与するところであり，対話システムも例外ではない．ユーザやシステムの発話中の語彙を Wikipedia 記事に結びつけることができれば，記事内容を対話システムの発話生成に利用できる．

対話システムとは異なるが，Wikification を会話データに行う研究がある．Kim ら [6] は，シンガポールにおける観光者とガイドの会話データに対し SVM と RankingSVM による学習を行い，発話中の名詞句をメンションとした Wikification を行っている．

本研究では，Twitter 会話データをコーパスの構築に利用するが，Twitter データを Wikipedia の記事に結びつける研究は，多量の Tweet（Twitter データ 1 つに相当する．Tweet1 つの文字数は 140 文字以内である.）をまとめて Wikipedia の記事に結びつけるもの，Tweet 一つ一つを Wikipedia の記事に結びつけるものに分けることができる．

前者の観点で，Twitter データを Wikipedia と結びつける研究としては [7] がある．Yıldırım ら [7] は，TF-IDF を利用し，2012 年にオバマ大統領が遊説した際の多量の Tweet を 2 分単位で区切り，2 分ごとにトピックを表す Wikipedia 記事に結びつけている．

後者の観点で，Twitter データを Wikipedia と結びつける研究には [8] や [9, 10] がある．短

文は，長文と比較した際に利用できる情報の少なさから困難が伴う．Genc ら [8] は，Tweet 間の類似度を Tweet を Wikipedia 記事に結びつけることで導出している．2 つの Tweet の類似度を測る際に，2 つの Tweet を Wikipedia 記事にそれぞれ結びつけ，結びつけられた 2 つの Wikipedia 記事の距離を 2 つの Tweet の類似度とする手法である．この 2 つの Wikipedia 記事の距離は，この 2 記事とカテゴリ記事のネットワークにおける 2 記事間のホップ数で表される．Ferragina ら [9, 10] は，Tweet のような短文中の語彙を Wikipedia 記事にリンクさせる手法を提案しており，これは Wikification の一種である．Ferragina らはアンカーテキストを利用したメンション同定とエンティティリンキングを行っており，その際同短文中の他のメンションとの関連性についても考慮し，Wikipedia 記事にリンクさせている．

2.2 対話システムにおける話題の考慮

雑談対話システムにおいて何を話題とするかは，研究によって異なるところである．Yoshino ら [11] が作成した対話システムを通してニュース記事を読むシステムでは，CRF を用いユーザが興味を持つ話題を同定している．これは述語項単位でその述語項が話題であるかを同定するものであり，この CRF には素性として述語項構造や品詞など 9 つの素性が用いられている．また，ニュース記事を読む上で，ユーザが最も求める情報を特定した利用をするため，1 発話中の話題は 1 つとした学習を行っている．Yoshino らの話題同定方法は，傾聴対話システムの研究においても利用されている [12].

功刀ら [13] による Yahoo!キーフレーズ抽出 API を利用する手法がある．功刀らは，ユーザの直前の発話からキーフレーズ抽出 API によって重要度の高いキーフレーズを抽出し，これを対話システムの応答作成時にコンテキストとして利用している．

対話システムの話題に Wikipedia を利用する研究には Wilcock ら [14] による WikiTalk がある．これは Wikipedia を読むことを目的とする，話題の幅に制限がない（オープンドメイン）対話システムであり，ユーザの反応に合わせて読む Wikipedia 記事を変更する機能を持つ．

2.3 対話システムのためのコーパス構築

非タスク指向型対話システムでの利用に向けたコーパス構築に関する研究には，[15]，[16] や [17] がある．別所ら [15] は話題継続の判定に向けたアノテーションデータの作成に取り組んでいる．別所らの研究では，システムとユーザが 1 度ずつ発話することをターンと呼ぶ．NTT の雑談対話システムにおけるシステムとユーザの会話データ 817 件の各ターンについて，

0. 直前のシステム発話の話題を継続したくない

1. 直前のシステム発話の話題を継続してもさしつかえない

2. 直前のシステム発話の話題を継続したい

という 3 段階の評価を，2 人のアノテーターに付与させている．

東中ら [16] は対話破綻検出のために，NTT ドコモによる雑談対話 API とユーザとの対話計 1,146 対話に対し

◯ 破綻ではない

△ 破綻と言い切れないが，違和感を感じる発話

× あきらかにおかしいと思う発話．破綻

というアノテーションを行っている．

Dinan ら [17] は，オープンドメインな対話において，知識ベースとして Wikipedia を利用するためのデータセットを作成している．2 人の話者を Apprentice と Wizard とに分け，どちらかが選択したトピック（Wikipedia の 1 記事に対応する）について 2 者は会話を開始する．Apprentice にはトピックについて熱心に知ろうとするような発話をするよう指示する．Wizard には，Apprentice とそのトピックについて熱心に語るよう指示する．この際 Wizard には，7 つの Wikipedia 記事の第 1 パラグラフ（直前の Apprentice と Wizard による発話，TF-IDF，bag-of-words，n-gram に基づき推薦される）と，会話開始時のトピックを表す Wikipedia 記事の先頭 10 文が提示され，Wizard はこれらに基づいた返答をするよう指示される．2 者間のトピックが変化していくことは許可されており，ひとつの会話は 2 者合わせて 10 以上の発話からなる．また，トピックの選定にはクラウドソーシングを用いている．

Dinan ら [17] に限らず，クラウドソーシングをコーパス構築に利用する試みはいくつか存在しており，構築されるコーパスは多岐にわたる．例えば，Potthast[18] は Wikipedia の荒らしを検出するコーパスの構築に，Filatova[19] は Amazon の商品レビュー中の皮肉を判別するコーパスの構築にクラウドソーシングを利用している．

非タスク指向型対話システムでの利用に向けたコーパスを構築にクラウドソーシングを利用する研究には [20] や [21] がある．塚原ら [20] は，雑談対話システムでの利用に向けたコーパスを開発のため，クラウドソーシングで集めたワーカー同士に雑談を行わせ，同時にワーカーには自身の発話に対話行為やトピックなどのアノテーションを行わせている．河原ら [21] はクラウドソーシングを用い談話関係のタグ付けコーパスを構築している．

2.4 研究の位置づけ

本研究と特に関連する研究は Dinan ら [17] の研究である．雑談対話においては，1 発話から派生する話題は複数ありえる．その点から本研究では，これを想定したコーパスを構築

する．Dinan らは，Wizard に 7 つの Wikipedia 記事の第 1 パラグラフを見せトピックの変化を認めた会話データを作成しており，これは本研究の目的と共通している．

また，Dinan らが具体的なトピックから始まる会話を想定している一方で，本研究で利用する Twitter 会話データには，例えば挨拶のみで終了する会話など，具体的なトピックを伴わない会話が存在する．加えて，Dinan らは 10 以上の発話により 1 会話が構成される会話データを収集しているが，Twitter 会話データにおいては，計 3 発話で終了する会話が 41.2%を占める（3.2 節）．実際のテキスト会話データである Twitter 会話データを利用し，話題がない会話まで対象にアノテーションを行う点が [17] と比較した際に本研究が持つ大きな差異である．

第3章　コーパスの構築手法

この章では，本研究で提案する非タスク指向型対話システムでの話題同定のためのコーパスの構築方法を述べる．

1章でも述べたように，本研究では，会話中の発話を“会話している人らにとって有益な記述”が存在するWikipedia記事に結びつけるコーパスを構築する．これを具体的にした以下の基準に基づくアノテーションを会話データ（3.2節）に行い，2であると判断する場合にはそのWikipedia記事中の有益な記述の収集を行う．

2. 会話している当人たちにとって有益な記述が記事中にある

1. 見出し文から，会話している当人たちにとって有益な記述があってもおかしくないと思われるが，現時点の記事中にはない

0. 見出し文からその記事中に，会話している当人たちにとって有益な記述がないとわかる

ところで，会話での話題がどのように決まるか，変化していくかについて決まりはなく，何を会話の話題とするかは人によって異なる．また，何が有益であるかの判断についても人により異なると仮定する場合，アノテーター1人のみにより構築されるコーパスは客観性に欠けたものとなる．そのため，複数人によるアノテーションを行い，アノテーターが2と判断する都度，記事中の有益な記述も収集する必要がある．そのため本研究では

ステップ1

1人のアノテーターが発話から抽出されたWikipedia記事に0と1をアノテーションするステップ

ステップ2

ステップ1でアノテーターが1とアノテーションしたWikipedia記事中に会話にとって有益な記述が存在するか判断と，有益な記述が存在する場合には有益であると判断した記述の抽出を，クラウドソーシングによって行うステップ

ステップ3

再度クラウドソーシングを利用し，有益であると判断された記述についてどれだけの合意が得られるかを確認するステップ

の 3 つのステップによりコーパスの構築を行う．1 章で述べた 2 段階のクラウドソーシングは，ステップ 2，ステップ 3 に対応する．

この章では，本研究で構築するコーパスの仕様について述べ，このコーパスを構築するため，会話データとその前処理，ステップ 1，ステップ 2，ステップ 3 について，順に詳細を述べる．

3.1 コーパスの仕様

本研究で構築するコーパスは以下から構成される．

- ステップ 1 でアノテーションしたテキストデータ 1,000 件（conv_0.txt ～ conv_999.txt）
- ステップ 2 でのアノテーションを集計した辞書型データの json ファイル
- ステップ 3 でのアノテーションを集計した辞書型データの json ファイル

この節では，上記についての詳細を具体例を用い，順に述べる．

ステップ 1 でアノテーションしたテキストデータ 1,000 件

本研究では，コーパス構築に 1,000 件の Twitter 会話データ（3.2 節で詳述）を利用する．ステップ 1 でアノテーションした各テキストデータは，元の Tweet（発話 1 つに相当）を表す行，最長マッチによって元の Tweet から抽出した記事名をタブ区切りで表した行，各記事名にアノテーターが 0 と 1 のアノテーションを行いタブ区切りで表した行の繰り返しにより構成され，1 つの会話が 1 つのテキストデータに対応する．記事名を表す行とアノテーション結果を表す行は，タブ区切りにした際の要素数が等しい．

例えば，conv_492.txt が

```
やっと、じゃがりこ食べたい症候群治った＼(^^)／笑
や っ と 読点 じゃがりこ 食 べ 鯛 症候群 た バックスラッシュ 括弧 サーカムフレックス スラッシュ（記号）笑い
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
まぢか（笑）　じゃがりこだいすき ww
ま ぢ か （笑） じゃがりこ
0 0 0 0 1
うん笑　じゃがりこ食べたもん＼(^^)／
う ん 笑い じゃがりこ 食 べ バックスラッシュ 括弧 サーカムフレックス スラッシュ（記号）
0 0 0 1 0 0 0 0 0 0
まぢか（笑）
ま ぢ か （笑）
0 0 0 0
まぢまぢ笑
ま ぢ 笑い
0 0 0
そっか（笑）
そ っ か （笑）
0 0 0 0
```

であるとすると，これは会話データ 492 の第 1 発話から第 3 発話の “じゃがりこ” それぞれにアノテーターが 1 とアノテーションしたことを表す．なお，このコーパスでは発話を 1 から数える．

ステップ 2 でのアノテーションを集計した辞書型データの json ファイル

ステップ 2 のアノテーションを集計した json ファイルは，各発話番号と発話中で 1 が出現した発話番号とを “_” で繋いだ文字列，1 とした記事名を key とし，集計結果をタプルにしたものを value とする辞書型データを json ファイルにしたものである．集計結果のタプルの要素は，有益であるとしたアノテーター（ステップ 2，ステップ 3 のアノテーターを本研究ではランサーと呼ぶ．3.4 節参照）の数と有益ではないとしたランサーの数を数える辞書，この辞書で多数決を行った場合にどちらが多数派であるかを表すものからなる．

例えば，

```
'492_3': {' じゃがりこ': ({' 有益である': 4, ' 有益ではない': 1}, ' 有益である')}
```

は conv_492.txt の第 3 発話中の著者がステップ 1 で 1 とアノテーションした記事 “じゃがりこ” に対し，有益であるとしたランサーが 4 人，有益ではないとした人数が 1 人であったことを表し，多数決を行う場合 “有益である” と判断されることを表す．

ステップ 3 でのアノテーションを集計した辞書型データの json ファイル

ステップ 3 のアノテーションを集計した json ファイルは，各発話番号と発話中で 1 が出現した発話番号とを “_” で繋いだ文字列，ステップ 2 でランサーが有益だとした記事名，有益だとしたランサーの名前を key とし，このランサーが Wikipedia から抽出した記述と集計結果からなるタプルを value とする辞書型データを json ファイルにしたものである．

例えば，

```
'492_3':{' じゃがりこ':{'xxxx': (' じゃがりこは、1995 年（平成 7 年）からカルビー株式会社が
製造・販売しているジャガイモを主原料としたスナック菓子で、同社の登録商標（第 4387394 号ほか）
である。一度ふかしたジャガイモを細い棒状に整形して油で揚げたもので、カップ状の容器に入れられ
ている',{' 有益である': 1, ' 有益ではない': 2},' 有益ではない'),
                                    ⋮
                        }
        }
```

は conv_492.txt の第 3 発話のステップ 2 でランサー “xxxx” は記事 “じゃがりこ” が有益であるとし，その際 “xxxx” が抽出した “じゃがりこは、1995 年” から始まる記述は，ステップ 3 において 1 人が有益であるとし，2 人が有益ではないとしており，多数決を行う場合は “有益ではない” と判定されることを表す．この集計結果からはスパム（3.5.1 節で詳述）が取り除かれている．なお，この例での “xxxx” はダミーであり，実データでのランサーの名前とは異なっている．また，ステップ 2 で作成する json ファイルの例を引き継ぐと，conv_492.txt の第 3 発話の記事 “じゃがりこ” をステップ 2 で有益であるとしたランサーは計 4 人存在するため，“xxxx” 以外の 3 人が抽出した記述に対しても “xxxx” と同様の集計が，ステップ 3 で作成する辞書の “492_3” の “じゃがりこ” 下に含まれる．

3.2 Twitter 会話データの前処理

本研究では，2013 年 4 月 1 日から 2013 年 6 月 3 日の Twitter 会話データを利用し，コーパスを構築する．本会話データにおいては，1 発話はユーザ 1 人の 1Tweet に対応し，1 会話は 2 人のユーザが交互に繰り返す発話からなる．この会話データは計 2,816,666 件の会話を集めたもので，各会話は 3 以上の発話（2 人の発話合わせて）により構成されている．発話回数ごとの集計を表 3.1 に記す．発話数 3 のデータは 1,159,132 件であり，全体の 41.2%を占めている．

この Twitter 会話データから 1,000 件の会話データをランダムに抽出し，1,000 件の会話データ中の各発話からアノテーションの候補となる Wikipedia 記事名を抽出する．発話中のハッシュタグと URL については，記事名の抽出前にあらかじめ取り除く．

記事名の抽出には最長マッチを利用する．最長マッチには，2017 年 6 月 20 日の日本語版 Wikipedia のダンプデータから作成する Wikipedia の記事名一覧を利用する．Twitter 会話データが 2013 年のものであることから，話題候補となる記事名はカバーできると判断するためである．また，リダイレクト記事 (付録 A) は，リダイレクト先の記事名として抽出することとする．なお，同じ発話から同じ記事が 2 回以上抽出可能な場合，1 回のみ抽出することとする．

例えば，

"うん笑　じゃがりこ食べたもん＼(^^)／"

という発話からは

{ “う”, “ん”, “笑い”, “じゃがりこ”, “食”, “べ”, “タモ”,
“バックスラッシュ”, “括弧”, “サーカムフレックス”, “スラッシュ（記号）” }

という記事名が抽出される．記事 “^” は記事 “サーカムフレックス” へのリダイレクト記事であるため，発話中の “^” から記事 “サーカムフレックス” が抽出される．また，発話中に “^” は 2 回出現しているため，記事 “サーカムフレックス” は 2 回抽出可能であるが，1 回のみ抽出している．

その後，抽出した各記事のうち，曖昧さ回避ページに該当する記事を取り除く．曖昧さ回避ページはリンクが列挙される記事であり，特定の内容を記述する記事ではないため，話題同定のためのコーパスには利用できないためである．曖昧さ回避ページは，ダンプデータ中の該当記事の “ダンプデータにおける本文”（付録 A 参照）内の曖昧さ回避ページを表すテンプレート（“{{” と “}}” で囲まれた文字列）の有無によって判別が可能である．記事 “Wikipedia:曖昧さ回避”[1] に基づいた正規表現により，記事の曖昧さ回避ページを識別する．

[1] https://ja.wikipedia.org/wiki/Wikipedia:曖昧さ回避

{ “う”, “ん”, “笑い”, “じゃがりこ”, “食”, “べ”, “タモ”,
“バックスラッシュ”, “括弧”, “サーカムフレックス”, “スラッシュ（記号）” }

から曖昧さ回避ページを取り除くと，記事 “タモ” が取り除かれ，以下のようになる．

{ “う”, “ん”, “笑い”, “じゃがりこ”, “食”, “べ”,
“バックスラッシュ”, “括弧”, “サーカムフレックス”, “スラッシュ（記号）” }

なお，曖昧さ回避ページである記事 “タモ” には，記事 “トネリコ”，記事 “ヤチダモ”，記事 “タモ網（たもあみ）” へのリンクが列挙されている．

このような前処理を行なった結果，1,000 件の会話データから，58,864 件の Wikipedia 記事名が，アノテーション候補として抽出された．

3.3 ステップ 1：0 と 1 のアノテーション

前節で 1,000 件の会話データから抽出した 58,864 件の Wikipedia 記事名に，0 もしくは 1 をアノテーションする．0 と 1 のアノテーションの基準を以下に再掲する．

1. 見出し文から，会話している当人たちにとって有益な記述があってもおかしくないと思われるが，現時点の記事中にはない

0. 見出し文からその記事中に，会話している当人たちにとって有益な記述がないとわかる

本研究では，ステップ 1 のアノテーションを著者自らが行う．各会話データについて，第 1 発話から順にアノテーションを行うが，その際，第 1 発話からアノテーションを行う発話までの会話（第 1 発話へのアノテーションの際は第 1 発話のみ）と，アノテーションを行う発話から抽出した記事の見出し文（付録 A 参照）のみを確認しアノテーションを行うこととする．これは，雑談における話題は複数存在しうることを想定したコーパス構築のためには，続く会話の影響を受けるべきではないためである．

このステップを著者が行う理由は 3 つある．1 つ目は，抽出した Wikipedia 記事名の中には “っ” や “長音符” など，明らかに話題とはならない記事名が多く含まれるためである．このステップでのアノテーション結果は以降のステップでの土台となる．そのため，このステップでクラウドソーシングを利用し，明らかに話題とはならない記事名が，例えば，ランダムにアノテーションするアノテーターや常に 1 とアノテーションするアノテーターといった不真面目なアノテーターによって，1 とアノテーションされる事態は避けるべきである．

2 つ目は，コストの問題である．最長マッチによって記事名抽出を行う都合，上記のような明らかに話題とはならない記事名を含め多くの記事名が抽出される．記事名単位でのアノテーションをタスクとして設計する場合にタスク数が膨れ上がる一方で，発話単位でのアノ

テーションをタスクとして設計するには，1 タスクでアノテーションを行う記事名の増加が乱雑なアノテーションの増加へとつながる懸念がある．

3 つ目は，個人の特定性が高い会話が含まれるためである．Twitter での会話には，LINE や Skype などの ID や所属する組織名など特定性の高い発話がしばしば含まれる．クラウドソーシングを利用するにはこれらは不適である．

このステップで見出し文を利用する最大の理由は，次ステップへの移行を早めるためである．アノテーション候補となる Wikipedia 記事すべての本文をアノテーター 1 人が読むには時間がかかる．また，Wikipedia 記事には，見出し文だけで明らかに話題ではないと判断できる記事が多く存在する（例えば “っ” や “長音符” は多くの会話に出現するが明らかに話題でないことが多い）．加えて，Wikipedia 記事の本文を見る作業は次ステップに任せること，経験的に，見出し文が表す内容が大きく書き換えられることは少ないと判断することからも，このステップでは見出し文を利用することとする．

また，アノテーションの際は，見出し文ではなく記事名だけを見て思い込みで判断しないように注意する必要がある．記事名を見るだけでは誤ったアノテーションをしうる例として，

> “うちはおひとりさまコースを歩みそうです…転校続きで身長伸びんかったし。”

という発話を挙げる．この発話から最長マッチによって記事名を抽出すると，Wikipedia 記事 “おひとりさま” を抽出できる．記事 “おひとりさま” が一般的な意味のおひとりさま [2] を表すのであれば 1 とアノテーションする候補となりうる．しかし，記事 “おひとりさま” のダンプデータにおける見出し文は，

> 『’’’おひとりさま’’’』は、TBS テレビ Japan News Network の金曜ドラマ（TBS）枠（毎週金曜日 22:00 - 22:54、日本標準時）で 2009 年 10 月 16 日から 12 月 18 日まで放送された日本のテレビドラマ。制作プロダクションはメディアミックス・ジャパン。主演は観月ありさ。 キャッチコピーは「’’’ で、何が悪いのよ!’’’」。

である（ただし，アンカーはリンク先の記事名に置き換えてある）．これは発話中の “おひとりさま” とは異なる “おひとりさま” について記した記事であるため 0 である．上記に注意し，著者が 1,000 件の会話データにアノテーションをしたところ，ひとつでも Wikipedia 記事に 1 をつけた会話データは 345 件となった．この 345 件の会話データ中に，1 をつけた Wikipedia 記事が 811 件存在する．

また，Twitter 会話データのうち，個人情報を含む会話など個人の特定性が高い会話は，クラウドソーシングに利用するには不適である．ステップ 2 ではクラウドソーシングを利用するため，1,000 件の会話データについてアノテーション行いながら，著者がどのようなアノテーションをするかに関わらず，個人名やニックネームを含む会話，LINE や Skype など

[2] https://www.weblio.jp/content/おひとりさま

の ID を含む会話，学校名など所属する組織を特定できる会話であるかを確かめ，ステップ 2 の対象から確実に取り除くこととする．結果，1,000 件の会話データのうち，573 件の会話データが残った．この 573 件の会話データのうち，ひとつでも Wikipedia 記事に 1 をつけた会話データは 202 件であり，この 202 件の会話データ中に，1 をつけた Wikipedia 記事が 481 件存在する．

3.4 ステップ 2：クラウドソーシングを用いた 1 と 2 のアノテーション

このステップの目的は，前節で著者が 1 とアノテーションした Wikipedia 記事が，

2. 会話している当人たちにとって有益な記述が記事中にある

1. 見出し文から，会話している当人たちにとって有益な記述があってもおかしくないと思われるが，現時点の記事中にはない

のどちらを満たすか判別するとともに，2 を満たす場合には有益である記述を収集することである．また，有益であるかどうかは人により異なるところであり，有益だと判断した場合についても，有益であるとした根拠となる記述が異なることもありうる．

そこで，このステップではクラウドソーシングサービス Lancers[3] を利用する．タスク 1 つにつき，前節で著者が 1 と判断した Wikipedia 記事を含む発話と，その Wikipedia 記事へのリンク，加えて，直前の 4 発話まで（有益であるか判断をお願いする Wikipedia 記事を含む発話が第 1 発話である場合は第 1 発話のみを，有益であるか判断をお願いする Wikipedia 記事を含む発話が第 2 発話から第 4 発話までの発話である場合は，第 1 発話からアノテーションを依頼する発話まで）をワーカー（Lancers では，ランサーと呼ばれる）に提示する．なお，提示する会話については，A さんの発話から始まる，A さんと B さんの 2 者による会話であるとして提示する．そして，会話する 2 者にとって有益な記述が，提示する Wikipedia 記事中に存在するか判断するよう指示する．また，有益であると判断する際には有益だと判断した記述のコピーアンドペーストをするように指示することで，有益な記述の回収を行う．

なお，有益な記述について具体的な指示はしない．人によって異なる判断基準に依る，様々な有益な記述を回収するためである．また，上記のばらつきを想定するため，有益であるか判断をお願いする Wikipedia 記事ひとつにつき，5 人から回答を得ることとする．ランサーには，図 3.1 のように，記事 “二日酔い” に関する具体例をもとに，タスクについての説明を行う．

また，個人の特定性が高い会話以外にも，クラウドソーシングに利用するには不適な会話が存在する．そのため，クラウドソーシングを利用する前に，ステップ 1 で 1 をアノテーションした会話データのうち，ランサーに提示する発話と直前の 4 発話までを再度確認する．そ

[3] https://www.lancers.jp/

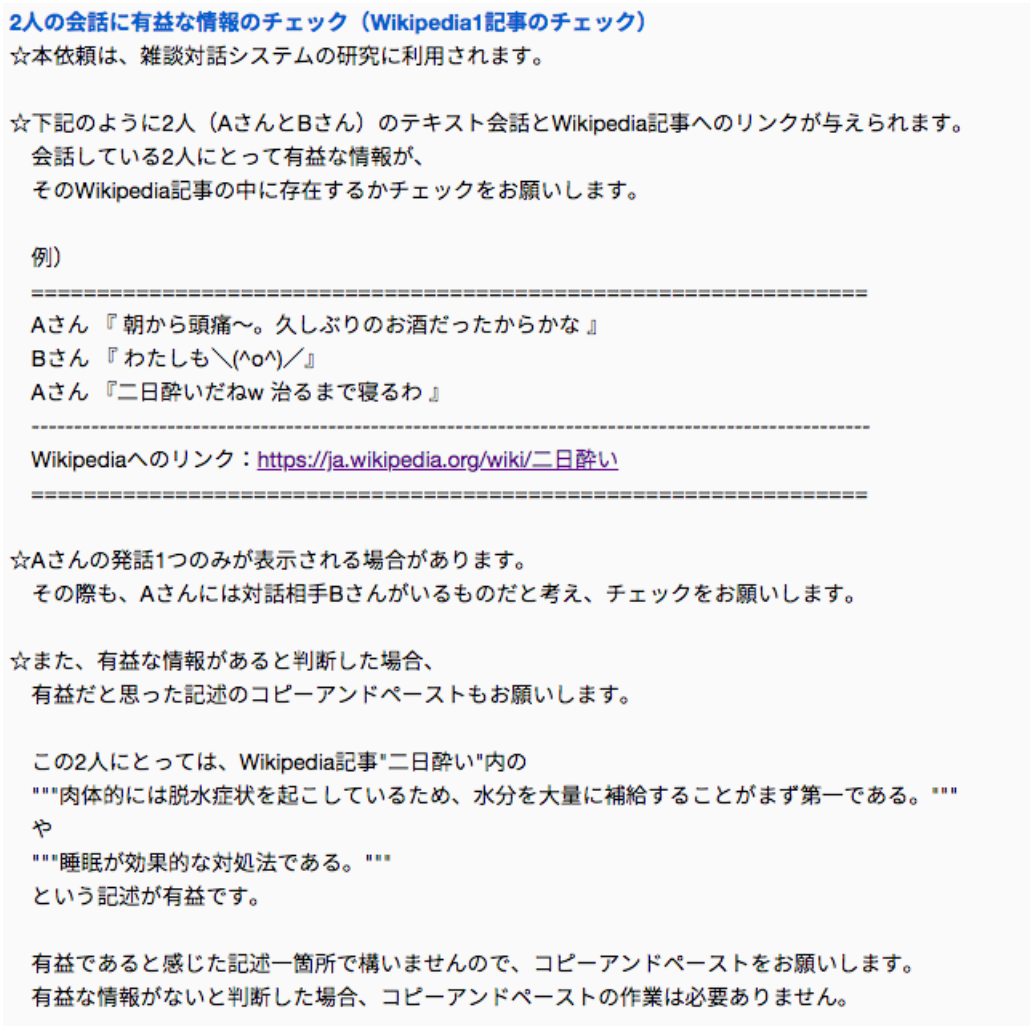

図 3.1: ステップ 2 でのランサーへのタスクの説明

のうち，政治的な発話を含む会話，悪口など他者を不快にする発話を含む会話，性的な発話を含む会話，ハッシュタグや URL を取り除いた結果空となった発話が存在する会話，“RT” や “フォローしました” といった Twitter の文脈知識を必要とする会話をクラウドソーシングに利用するには不適であるとして，取り除く．上記を満たす会話を取り除いた結果，ステップ 1 で獲得した，ひとつでも Wikipedia 記事に 1 をつけた会話データは 202 件のうち，153 件の会話データが残った．これにより，1 をつけた Wikipedia 記事 481 件のうち，370 件の 1 をつけた Wikipedia 記事が残り，ステップ 2 ではこの 370 件について回答を得ることとなる．1 をつけた記事ごとに 5 人から回答を得るため，得られる回答数は 1,850 件である．

ステップ 2 を行った結果，このステップに参加したランサーは 67 人であり，1,850 件の回答のうち，有益であるとの回答が 681 件であった．また，これに伴い，ランサーが有益だと判断した記述 681 件を獲得した．

3.5　ステップ 3：Lancers を用いた再度のアノテーション

このステップの目的は，ステップ 2 でランサーが有益だと判断した記述 681 件について，どれだけの合意が得られるか確認することである．そのために再度 Lancers を利用する．タスク 1 つにつき，有益だと判断された記述 1 件，その記述を含む Wikipedia 記事名，ステップ 2 と同様，その Wikipedia 記事名を含む発話とその発話までの最大 4 発話までを A さんと B さんによる会話としてランサーに提示する．ただし，ここで提示する有益だと判断された記述については，Lancers に利用できるデータサイズの都合，元の有益だと判断された記述を “。” と改行区切りで 1 文とした際の先頭 5 文までとする．そして，その Wikipedia 記事名に関連する記述であることを念頭に置き，ステップ 2 で有益だと判断された記述が，有益であるか判断するよう指示する．

ところで，ステップ 2 で有益であると判断された記事が漫画や人物などの固有表現を表す場合には，一般的な記事名の場合と比較して，記事名の説明に該当する記述がしばしば抽出されていた．これは，ランサーが記事名が表すものを知らないために，会話する 2 者にとって有益な記述ではなく，ランサーにとって有益な記述が抽出されている可能性を示唆している．そのため，ステップ 3 では “詳しく知っていた”，“知っていた”，“知らなかった” の 3 尺度で Wikipedia 記事名が表すものをあらかじめ知っていたかを問い，ランサーの事前知識が有益であると判断する際に影響するかを確認する．加えて，A さんや B さんが Wikipedia 記事名について知っていると思うかについても問う．これは 3.5.1 節で詳述する．

ステップ 3 では，ステップ 2 でランサーが有益だと判断した記述 681 件について，記述 1 つにつき 5 人から回答を得る．そのため，計 3,405 件の回答を得ることとなる．ランサーには，図 3.2 のように，記事 “二日酔い” に関する具体例をもとに，タスクについての説明を行う．

3.5.1　スパムの判定

ステップ 3 は，ステップ 2 でランサーが有益であるとした記述についての合意を調べることで，有益な記述の信頼性を高めるステップであることから，クラウドソーシング終了後にスパムであるランサーの回答を取り除くことでより信頼性を高める．本研究では，タスクに不真面目に取り組むランサーをスパムと定義し，B についての質問 “記事名があらわすものについて、B さんは知っていると思いますか？” の正解率によってスパムであるか判別をする．

A についての質問 “記事名があらわすものについて、A さんは知っていると思いますか？” では “詳しく知っていると思う”，“知っていると思う”，“知らないと思う”，“この会話だけでは判別できない” の 4 つの選択肢を用意するが，B についての質問 “記事名があらわすものについて、B さんは知っていると思いますか？” ではこれに “この会話に B さんは登場し

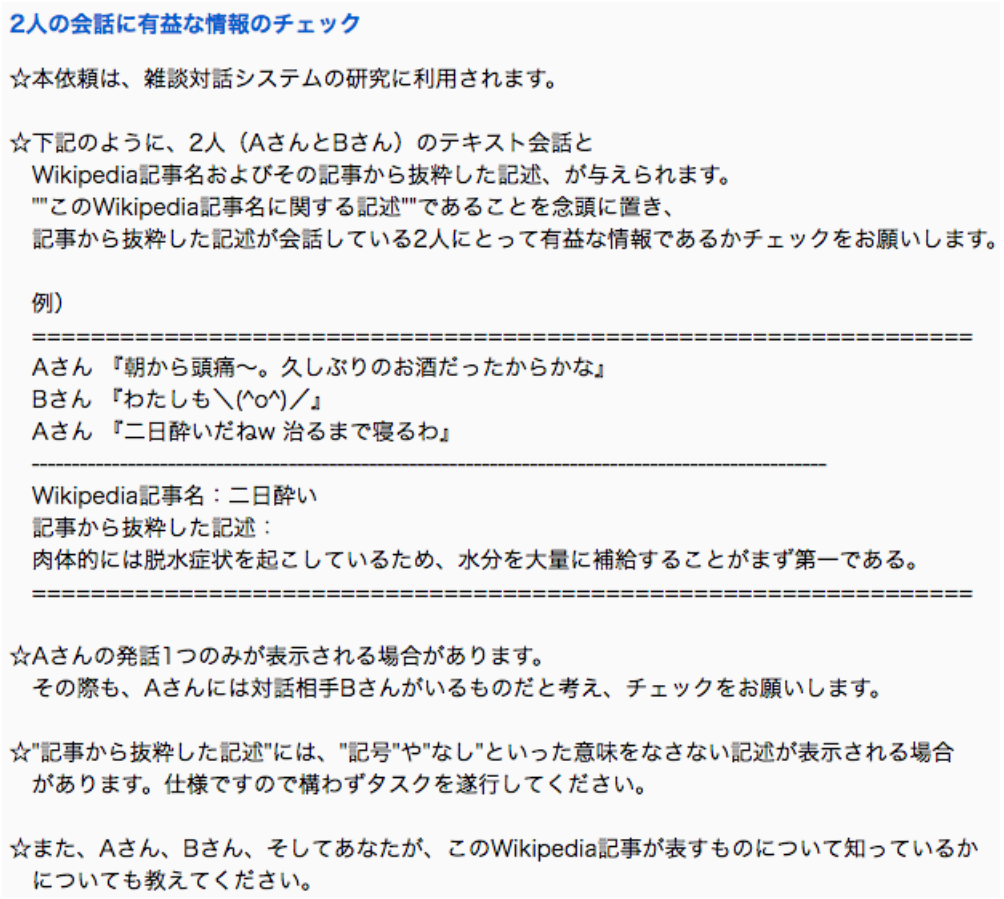

図 3.2: ステップ 3 でのランサーへのタスクの説明

ていない” の選択肢を加える．この選択肢によって，B についての質問は正誤を判定できる質問となる．それは，このクラウドソーシングでランサーに提示する会話データには，第 1 発話のみが表示される（A のみが登場する）会話データと，判定を依頼する発話と直前の最大 4 発話が表示される（A と B が登場する）会話データが存在するためである．A のみが登場する会話データに対しては，“この会話に B さんは登場していない” を選択した場合に正解とし，A と B が登場する会話データに対しては，“詳しく知っていると思う”，“知っていると思う”，“知らないと思う”，“この会話だけでは判別できない” のどれかを選択した場合に正解とする．これを用い，各ランサーについて正解率を算出し，スパム判定に利用する．

ただし，例えば，“唐揚げ” や “転勤”，“食欲” など一般的な記事名によっては，B が “知っていると思う” とのバイアスがかかることもありうる．加えて，図 3.2 に示したように，A の発話のみが表示される場合にも対話相手 B がいるものと想定したタスク遂行をするようアナウンスしていることから，A の発話内容によっては，“知らないと思う” や “この会話だけでは判別できない” とランサーが判断することもありうる．そのため，正解率 1.0 のランサー以外すべてをスパムとするわけにはいかない．

また，ここで，ステップ 3 で利用する 681 件の会話データ（ステップ 2 でランサーが有益だと判断した記述 681 件に伴う）について，A のみが登場する会話データと A と B が登場する会話データの件数は，それぞれ 262 件,419 件であり，比率はそれぞれ，0.385 と 0.615 であることを確認した．このことから，スパムであるランサーがこの確率に従いタスクを遂行し，かつ，例えば常に “詳しく知っていると思う” を選択すると仮定する場合，このランサーの正解率は 0.615 になる．このことから，正解率 0.615 未満のランサーをスパムではないと判断すると，スパムのタスク遂行結果が紛れ込むことが考えられる．

これらの観点から，経験的に正解率 0.85 未満のランサーをスパムと判定することとする．ただし，タスク遂行数 1 のランサーについては判別に利用できる情報が少ない都合，正解率に関係なくスパムではないとみなす．ステップ 3 には 80 人のランサーが参加し，うち 12 人がスパムと判定され，3,405 件の回答のうち 2,455 件が有効な回答として残った．

表 3.1: Twitter 会話データの発話数ごとのデータ数

発話数	データ数	発話数	データ数	発話数	データ数	発話数	データ数
3	1,159,132	45	223	87	17	130	3
4	516,209	46	241	88	14	131	6
5	339,615	47	188	89	20	132	5
6	218,602	48	191	90	11	134	1
7	147,039	49	197	91	20	135	2
8	104,410	50	138	92	7	136	2
9	72,536	51	156	93	3	137	2
10	54,893	52	153	94	3	138	1
11	39,094	53	122	95	14	139	2
12	31,511	54	125	96	9	140	3
13	23,026	55	113	97	15	143	1
14	18,715	56	86	98	5	145	4
15	14,238	57	90	99	6	146	2
16	12,127	58	77	100	9	148	2
17	9,220	59	78	101	6	150	5
18	8,050	60	74	102	6	151	2
19	6,304	61	58	103	4	153	1
20	5,615	62	60	104	5	155	1
21	4,433	63	60	105	2	156	1
22	3,945	64	63	106	1	158	1
23	3,241	65	47	107	9	161	1
24	2,879	66	44	108	8	162	2
25	2,309	67	36	109	2	163	1
26	2,152	68	41	110	3	165	1
27	1,821	69	56	111	5	169	1
28	1,569	70	25	112	7	175	1
29	1,394	71	27	113	6	181	1
30	1,253	72	25	114	2	188	1
31	1,018	73	24	115	4	189	1
32	969	74	35	116	2	194	1
33	770	75	31	117	5	195	1
34	781	76	37	118	2	197	1
35	644	77	24	119	3	200	1
36	632	78	18	120	1	207	1
37	552	79	20	121	1	212	1
38	506	80	15	122	2	216	1
39	424	81	21	123	1	227	1
40	408	82	14	124	2	234	1
41	334	83	12	125	1	238	1
42	337	84	13	127	1	242	1
43	306	85	11	128	2	282	1
44	257	86	11	129	1	**合計**	2,816,666

第4章　構築したコーパスの分析

この章では，提案手法に基づき構築されたコーパスについて，ステップ1，ステップ2，ステップ3の順に，具体例を確認しながら分析する．また，本研究で構築したコーパスを利用する上では，有益な記述を抽出する方法について検討する必要がある．そのため，4.4節では，TF-IDFベクトルのコサイン類似度を用いた手法によって本研究で獲得した有益な記述の抽出を試み，このシンプルな方法では有益な記述は抽出できないことを確認する．

4.1　ステップ1の結果の分析

1,000件の会話データおよび，ここから個人の特定性が高い会話を取り除いた結果残った計573件の会話データに対しステップ1に基づくアノテーションを行った結果は表4.1の通りである．表4.1より，0とアノテーションしたWikipedia記事名は全会話で出現したWikipedia記事名の98%以上を占めており，1とアノテーションしたWikipedia記事名に比べて極めて多い．

これは出現したWikipedia記事名の多くが話題とは関係がないことを意味している．表4.2に会話データ1,000件における出現回数上位30件の記事名とその出現回数を記す．表4.2に記す通り，最長マッチによってWikipedia記事名を抽出すると，ひらがな1文字や記号など，会話の話題とはなりにくい記事名が多く出現する．出現回数上位500件までの出現回数の累積を図4.1に示す．上位30件までの累積出現回数は28,917（全出現回数の49.1%），上位100件までの累積出現回数は43,832（全出現回数の74.5%），上位500件までの累積出現回数は53,233（全出現回数の90.4%）となる．なお，出現回数100位の記事の出現回数は89

表 4.1: ステップ1でのアノテーション結果

条件	該当データ数（1,000 会話）	該当データ数（573 会話）
第1発話での1の出現回数	302	194
全発話での1の出現回数	811	481
全発話での0の出現回数	58,053	27,416
全発話での記事の出現回数	58,864	27,897
出現した記事名の種類数	3,429	2,350
全発話数	5,253	2,678
1会話あたりの平均発話数	5.253	4.674
ひとつでも1をつけた会話の数	345	202
発話数最大の会話における発話数	46	27

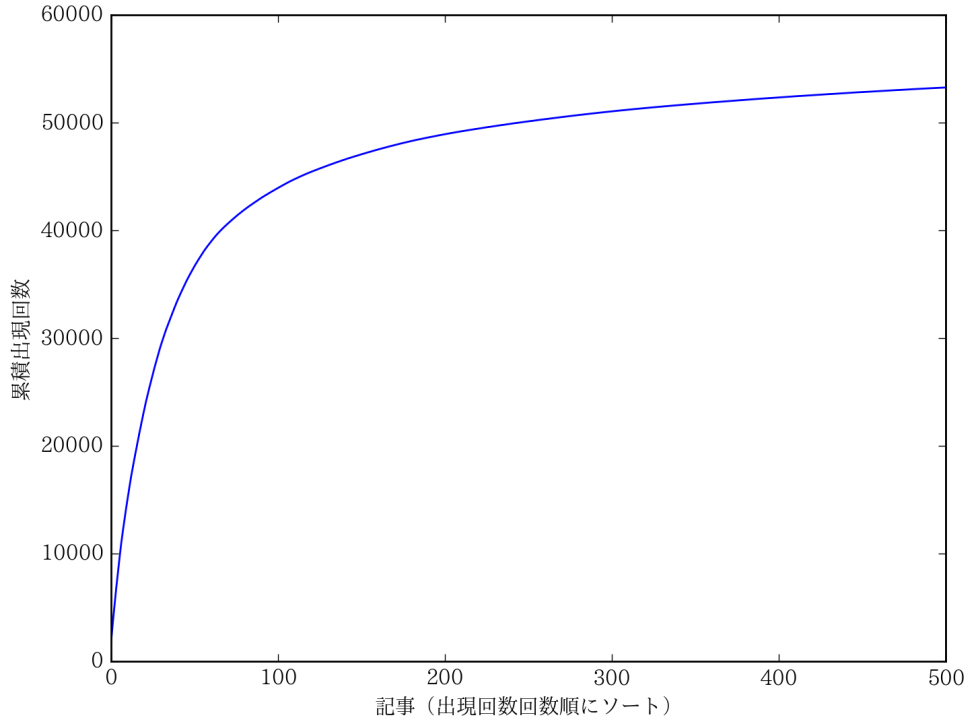

図 4.1: 会話データ（1,000 件）中の出現回数上位 500 件の記事の累積出現回数

回，出現回数 500 位の記事の出現回数は 8 回であった．

表 4.2: 会話データ (1,000 件) 中の出現回数上位 30 件の記事名と出現回数

記事名	出現回数	記事名	出現回数	記事名	出現回数
っ	1,839	だ	1,008	か	721
括弧	1,749	読点	982	が	707
な	1,576	た	953	す	640
て	1,564	る	850	く	631
で	1,433	も	834	や	611
い	1,397	句点	800	う	603
の	1,338	と	796	ま	599
長音符	1,169	は	793	お	558
ん	1,104	よ	763	れ	553
に	1,060	し	741	リーダー (記号)	545

4.2 ステップ 2 の結果の分析

ステップ 2 のクラウドソーシングは 2018 年 11 月 27 日 18 時 46 分から 2018 年 11 月 28 日 10 時 41 分の 15 時間 55 分で完了した．ステップ 2 のクラウドソーシングに利用した会話データ 153 件に対し，表 4.1 と同様の集計を行うと表 4.3 のようになる．

表 4.3: ステップ 2 のクラウドソーシングに利用した会話データの詳細

条件	該当データ数（153 会話）
第 1 発話での 1 の出現回数	147
全発話での 1 の出現回数	370
全発話での 0 の出現回数	9,843
全発話での記事の出現回数	10,213
出現した記事名の種類数	1,359
全発話数	818
1 会話あたりの平均発話数	5.346
ひとつでも 1 をつけた会話の数	153
発話数最大の会話における発話数	23

表 4.4: 有益である，有益ではないとしたランサーの数による Wikipedia 記事数および回答数の集計

有益であるとしたランサーの数	有益ではないとしたランサーの数	Wikipedia 記事数	有益であるとした回答数	有益ではないとした回答数
5	0	15	75	0
4	1	32	128	32
3	2	66	198	132
2	3	92	184	276
1	4	96	96	384
0	5	69	0	345
	合計	370	681	1,169

ステップ 2 に参加したランサーは 67 人であった．ステップ 2 で獲得した 1,850 件の回答を用い，“有益である” もしくは “有益ではない” としたランサーの数ごとに Wikipedia 記事数および回答数を集計した結果は，表 4.4 の通りである．84 件の Wikipedia 記事について，5 人全てのアノテーションが一致している．また，ステップ 2 で獲得した回答 1,850 件の内訳は，“有益である” が 681 件，“有益ではない” が 1,169 件であった．

図 4.2 は，ランサーをタスク遂行数順に並べた際のタスク遂行数と有益であるとした回数を表す．何を有益とするかの判断基準は人により異なっており，例えば，有益であると判断する回数はタスク遂行数に比例するなどの法則性がないことがわかる．図 4.3 は，ランサーのタスク遂行数と，タスク遂行数に占める有益であるとした回数の割合を表す．図中の赤線は，ランサーの有益であるとした割合の平均 0.482 を表す補助線である．図 4.3 からは，ステップ 2 に参加したランサー 67 人のうち，タスク遂行数が多いランサー（タスク遂行数上位の 6 人）には，有益であるとした割合の平均 0.482 を超えるランサーが存在しないことがわかる．

表 4.4 よりステップ 2 でクラウドソーシングに利用された 370 件の Wikipedia 記事それぞれについて多数決によって，有益であるか有益ではないか判定すると，257 件が有益ではない，113 件が有益であると判定される．しかし，多数決で有益ではないと判定されてしまうが，そのうち有益であるとしたランサーが抽出した記述の中には，有益な記述が含まれることがあった．例えば

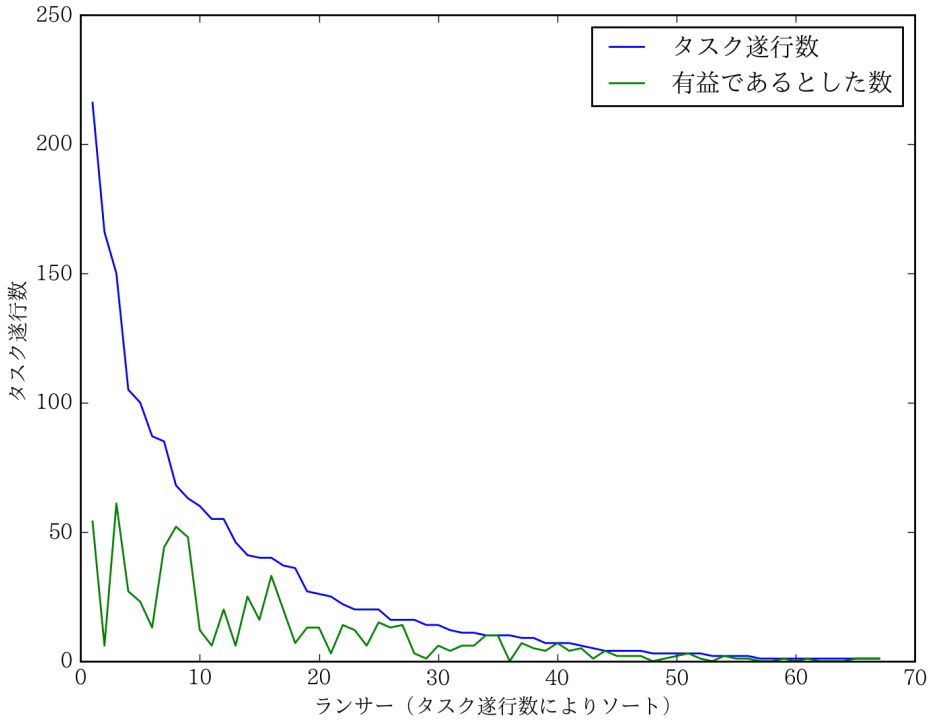

図 4.2: ランサーごとのタスク遂行数と，有益であるとした回数

> “A さん『そしたらひらめいた。ああ、私の諦めグセ、子どもの頃から何度もしてる転勤って、けっこう影響してるかも。と。親の転勤は、子ども心をいつも傷つける出来事を伴う。でもそれは、どうにも変えられないこと。諦めないといけないこと。幼稚園のとき、小学生の時、また転勤か、悲しいなってね。』”
> “B さん『転校は大変よな！うちも小学校の四回した！これからは自分が転勤族になりそう。』”

の 2 発話目からリダイレクト記事 “転校” のリダイレクト先として抽出される Wikipedia 記事 “転学” に，著者はステップ 1 で 1 をアノテーションしている．これに対してステップ 2 では，4 人が “有益ではない”，1 人が “有益である” であるとした．この有益であるとした 1 人が抽出した記述は，

> “イギリスのウォーリック大学が行った調査によると、12 歳より前の年齢で転校を経験した児童は、そうでない児童より 60%ほど精神障害になる可能性が高いという。研究者は、学校を移動すること自体が、幼い児童にとって大きな負担であり、精神疾患の原因になり得る可能性があるとしている [2]”

である．2 者は幼少時の転学の影響についての会話を行っているため，抽出された記述は十分に有益である．しかし，純粋な多数決によってこの記事が有益であるか判断をすると，“有益ではない” とされてしまう．

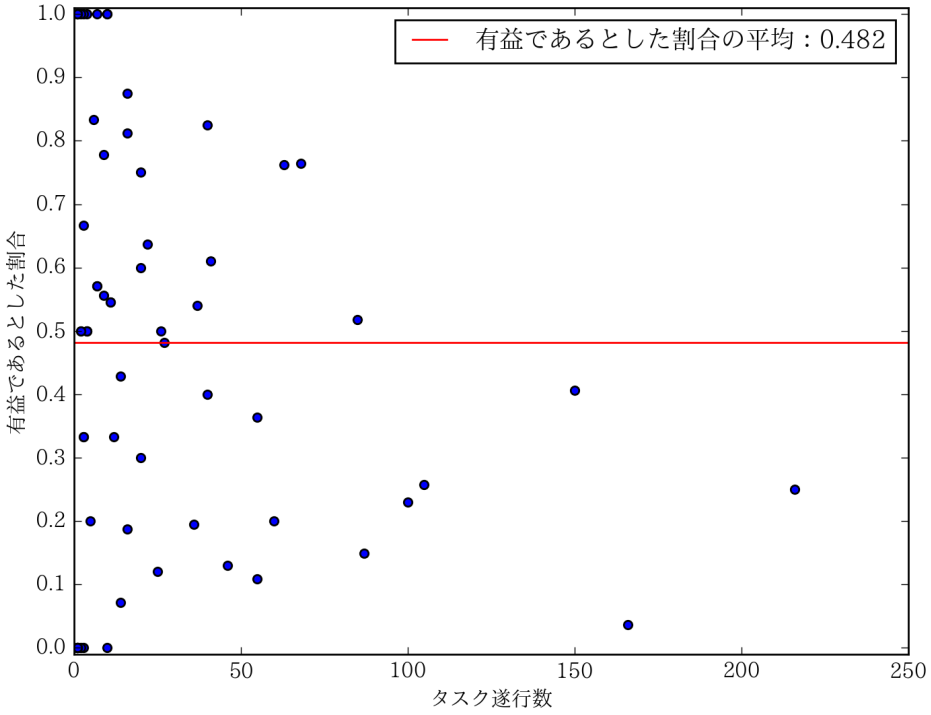

図 4.3: タスク遂行数と有益であるとした割合

また，有益であるとして抽出された記述の中には明らかに誤った記述が含まれることがあった．例えば

> “A さん『人生初のはもしゃぶとクジラのほっぺ食べて帰宅！明日は定休日
> 学校終わったら部屋の片付けと洗濯と洗い物全部する』”
> “B さん『クジラのほっぺてどこらへん？』”

の 2 発話目の記事名 “クジラ” を有益であると判断したあるランサーは，

> “肉体的には脱水症状を起こしているため、水分を大量に補給することがまず第一である”

を有益な記述であるとするタスク遂行結果を提出している．しかし，これは記事 “クジラ” 内の記述ではなく，図 3.1 で示したように，著者がタスクの説明時に提示した記事 “二日酔い” 内の記述であり，明らかに誤ったタスク遂行結果である．有益な記述が抽出されていない場合や，明らかに有益ではない記述が有益であるとして提出されることからも，記事単位で多数決を行い記事が有益であるかを判定することはできない．

ステップ 2 の結果の分析をまとめると以下の通りとなる．

- 何が有益であるかの判断基準は人によって異なる

表 4.5: ステップ 3 で有益である，有益ではないとしたランサーの数による，ステップ 2 で有益とされた記述の集計
（罫線上部が “過半数がステップ 3 で有益であるとした記述 389 件についての集計” を，罫線下部が “半数以下がステップ 3 で有益ではないとした記述 292 件についての集計” を表す）

ステップ 3 で **有益であるとしたランサーの数**	ステップ 3 で **有益ではないとしたランサーの数**	ステップ 2 で 有益であるとされた記述の数
5	0	14
4	1	13
4	0	62
3	2	19
3	1	102
3	0	66
2	1	94
2	0	18
1	0	1
1	4	9
0	4	13
2	3	8
1	3	43
0	3	20
2	2	109
1	2	67
0	2	3
1	1	20

- 多数決を行うと有益ではないとされてしまう記事であっても，有益であるとしたランサーによって有益な記述が抽出されている場合がある
- 明らかに有益ではない記述が，有益であるとして提出されることがある

4.3 ステップ 3 の結果の分析

ステップ 3 のクラウドソーシングは 2018 年 12 月 25 日 19 時 52 分から 2018 年 12 月 26 日 15 時 15 分の 19 時間 23 分で終了した．ステップ 3 に参加したランサーは 80 人であり，うち 12 人がスパムと判定され，68 人の回答が残った．ステップ 3 で得られた回答は 3,405 件であり，スパムの回答を取り除くと 2,455 件の回答が残った．この 2,455 件の回答を利用し，ステップ 3 での有益であるか，有益ではないかの回答数によって，ステップ 2 で獲得した有益な記述 681 件の集計を行った結果は，表 4.5 の通りとなる．681 件の有益な記述すべてに対し 1 人以上の回答が残った．表 4.5 に対し，有益であると判断したランサーが過半数を超えた場合をその記述が有益であるとする集計を行うと，389 件が有益であり，292 件が有益ではないという結果となる．

ステップ 3 で 5 人が有益だと判断した記述の例を挙げる．

“A さん『久々に筋トレしようとしたら腕立て 10 回ももたんくて萎えた (笑)』”
“B さん『今日は筋肉痛覚悟で頑張る。』”

の 2 発話目の記事 “筋肉痛” 中の記述

“痛みを和らげる方法としては、冷やす、時間がたってから安静にする・入浴などで筋肉を温めるといった「消極的休息」のほか、軽度の運動やストレッチングなどで血行をよくする「積極的休息」がある。伸張性収縮を極力起こさないように運動を行えば、筋肉痛を抑えることも出来る”

に対し，5 人が “有益である” との判断を下した．

4.2 節で挙げた Wikipedia 記事 “転学” 中の記述

“イギリスのウォーリック大学が行った調査によると、12 歳より前の年齢で転校を経験した児童は、そうでない児童より 60%ほど精神障害になる可能性が高いという。研究者は、学校を移動すること自体が、幼い児童にとって大きな負担であり、精神疾患の原因になり得る可能性があるとしている [2]”

については，スパム 2 人のアノテーションが削除され，2 人が有益であるとし, 1 人が有益ではないと判定する結果となった．

4.2 節の Wikipedia 記事 “クジラ” にはない明らかに誤った抽出がなされた記述

“肉体的には脱水症状を起こしているため、水分を大量に補給することがまず第一である”

については，1 人のスパムを取り除いた結果，残った 4 人全員が有益ではないと判断した．有益ではないのにステップ 2 では有益であるとされた記述が正しく有益ではないという判定を受けている．

一方，表 4.5 から，有益であると判断したランサーが 2 人，有益ではないと判断したランサーが 2 人という，票数が等しい記述が 109 件と最多であることも分かる．何を有益とするかの判断基準は人によって異なることを本研究では仮定しているが，この結果もそれを表していると言える．

また，同記事内の記述でも有益であるか有益ではないかの判断が大きく異なるものも存在した．会話

“A さん『卒業とか進級とかできないで焦る夢、いまだに見るよ、まじで』”

の 1 発話目の記事 “夢” 中の記述

“夢占い（あるいは夢判断）では、夢は見た者の将来に対する希望・願望を指すか、これから起き得る危機を知らせる信号と考えられている。また、夢でみた現象がそのまま実現する夢を予知夢と呼び、可能性がある夢を詳細に検討する場合もある”

表 4.6: ランサーがタスクに取り組む前から記事が表すものを知っていたかと，有益であるか有益ではないかの判断による回答の集計

	有益である	**有益ではない**	**合計**
詳しく知っていた	151	175	326
知っていた	1,099	575	1,674
知らなかった	279	176	455
合計	1,529	926	2,455

に対して，5 人が “有益である” との判断を下した．一方で，同発話中の同記事 “夢” 中の記述

> “夢とは何なのか？ということについては、古代からある信仰者の理解、20 世紀の心理学者の理解、現代の神経生理学者の理解、それぞれ大きく異なっている”

に対して，スパムを取り除いた計 4 人が “有益ではない” との判断を下している．このように，同じ記事から抽出した記述であっても有益であるか有益ではないかの判断が大きく異なるため，非タスク指向型対話システムの話題として Wikipedia を利用する際には，記事ではなく，記述の利用が重要であると言える．

また，ステップ 3 では，ランサーに対してタスクに取り組む前から該当する Wikipedia 記事名が表すものを知っていたかを質問している．これが有益であるかの判断に影響するか確認するために集計を行ったものが表 4.6 である．タスクに取り組む前から該当記事が表すものについて “知っていた”，“知らなかった” ランサーによる回答は，どちらも “有益である” の回答数が “有益ではない” の回答数よりも多い．一方で，“詳しく知っていた” ランサーによる回答は，“有益ではない” の回答数が “有益である” の回答数よりも多い．これは “知っていた”，“知らなかった” ランサーとは異なる傾向である．なお，知らなかったと判断するランサーが多数派となる記事名には，“スマイルプリキュア!”，“ハイキュー!!” などのアニメや漫画を表す記事名や “イングヴェイ・マルムスティーン”，“ヒュー・ジャックマン”，“一ノ関圭” といった人名などの固有表現が多く見られた．知っていたと判断された記事名には “筋肉痛”，“ジャガイモ”，“クジラ” などの一般的な記事名が多く含まれており，詳しく知っていたと判断される記事名にも同様の傾向が見られ，“モヤシ”，“歯ブラシ”，“風邪” など一般的な記事名が多く見られた．

ステップ 3 の結果の分析をまとめると以下の通りとなる．

- 何が有益であるかの判断基準は人によって異なる
- ステップ 2 で抽出された明らかに有益ではない記述が，ステップ 3 で有益ではないと判定された
- ステップ 2 で多数決を行うと有益ではないとされてしまう記事から抽出された記述にも，ステップ 3 で有益だと判断される記述が存在する

- 同じ発話から抽出された同じ記事内の記述でも，有益であるとするランサーの数が大きく異なる場合があるため，非タスク指向型対話システムの話題として Wikipedia を利用する際には，記事ではなく記述の利用が重要である
- ランサーが事前にその記事が表すものを詳しく知っている場合，記述が有益ではないと判断することが増える
- 知らなかったと回答するランサーが多数派となる記事には創作物や人名を表す記事などの固有表現が多く含まれていた

4.4 TF-IDF ベクトルを用いた有益な記述抽出の難しさ

本研究で構築したコーパスを利用する上で検討すべきことは，有益な記述を抽出する方法である．そこで，この節では，TF-IDF ベクトルのコサイン類似度によって，本研究で獲得した有益であると判断された記述の抽出を試みる．その後，本研究で獲得した有益であると判断された記述は，上記のシンプルな方法では抽出できない記述であることを確認する．そのためには，まずランサーが抽出した記述がダンプデータのどの文に対応するか同定する必要がある．ランサーが Web 上の Wikipedia から抽出した記述とダンプデータにおける本文とは表記が異なる部分が存在し，ダンプデータと完全一致する部分を抽出したとみなす方法では，ダンプデータにおける本文に当てはめることができないためである．この節では，ランサーが抽出した記述の同定方法を述べた後，この記述が TF-IDF ベクトルのコサイン類似度によるシンプルな方法では獲得できない記述であることを確認する．

4.4.1 抽出した箇所の同定

ランサーが有益であると判断した記述を “。” と改行で区切ったもの（それぞれ文とする）のリストを

$$L = [l_1, \ldots, l_j, \ldots, l_J]$$

とし（ただし，元の順を保持する），その記述が含まれる Wikipedia 記事の 2018 年 12 月 1 日の Wikipedia ダンプデータにおける本文を “。” と改行で区切ったもの（それぞれ文とする）のリストを

$$W = [w_1, \ldots, w_i, \ldots, w_I]$$

とする（ただし，元の順を保持する）．2018 年 12 月 1 日のダンプデータを利用するのは，ステップ 2 が行われた日時以降のダンプデータで，この日時に最も近いダンプデータが 2018 年 12 月 1 日のものであるためである．

Algorithm 1 ランサーが抽出した記述をダンプデータにおける本文への当てはめ

Input: L:ランサーが抽出した記述を “。” と改行で区切ったリスト（出現順）,
W:該当 Wikipedia 記事を “。” と改行で区切ったリスト（出現順）
Output: S:抽出した記述をダンプデータにおける本文への当てはめたリスト

```
S = []
for each l in L do
  s = (∞, [])
  for each w in W do
    d = 編集距離 (l, w)
    if d < s[0] then
      s = (d, [(セクション番号, セクションでの発話番号)])
    else if d == s[0] then
      s[1].append((セクション番号, セクションでの発話番号))
    end if
  end for
  S.append(s)
end for
return S
```

L をダンプデータにおける本文に当てはめたリストを C とし，これを作成するため，まずは Algorithm 1 により

$$S = [s_1, \ldots, s_j, \ldots, s_J]$$

を作成する．S は L の各文をそれぞれダンプデータの本文に当てはめた際にどのセクション（付録 A）の何番目の文に該当するかを抽出したものである．例えば,

$$S = [(2, [(3, 4), (4, 7)]), (5, [(3, 5), (4, 8), (6, 2)]), (0, [(4, 9)])]$$

のような出力がなされる．この例では，l_1 は 3 セクションの 4 文目と 4 セクションの 7 文目に当てはめられる（編集距離は 2 である）ことを表す.

この S を利用し，最も長い連鎖を作ることができる文のつながり（この例ではセクション 4 の 7，8，9 文目．セクションが途切れる場合は繋がっていないとみなす）をランサーは抽出したとみなし，セクションごとの文番号を該当記事全体での文番号に置き換えたものを

$$C = [c_1, \ldots, c_k, \ldots, c_K]$$

とする．なお，ダンプデータにおける本文と web 上での表記が完全一致しない都合，ランサーが抽出した文を正しく抽出できず，S の連鎖が L の長さを下回る場合があるため $K \leq J$ である.

4.4.2 TF-IDF ベクトルを用いた有益な記述の抽出

W の各要素に対して MeCab[1] を利用し，一般名詞，固有名詞，サ変接続の名詞を獲得し，これを語彙とする TF-IDF ベクトルのリスト

$$TFIDF_{wiki} = [tfidf_1, \ldots, tfidf_i, \ldots, tfidf_I]$$

を作成する．加えて，語彙と IDF 値を対応付けて保存した辞書 IDF_{wiki} を作成する．

第 1 発話からステップ 2 でランサーが有益であると判断した発話までのすべての発話を $conv$ とする．IDF_{wiki} を利用し，$conv$ 中の一般名詞，固有名詞，サ変接続の名詞（ただし，それぞれ IDF_{wiki} に含まれるもののみ）を語彙とした TF-IDF ベクトル $tfidf_{conv}$ を作成し，$TFIDF_{wiki}$ を $tfidf_{conv}$ とのコサイン類似度が高い順にソートした際の文番号リストを獲得する．この文番号リストを

$$W_{sort} = [w_{s\_1}, \ldots, w_{s\_i}, \ldots, w_{s\_I}]$$

とする．なお，W_{sort} にはコサイン類似度が 0 となった文番号も含まれている．

その後 W_{sort} を先頭から順に走査し，C に含まれる文番号が何番目に出現するかを確認し，C に含まれる文番号がひとつでも見つかった場合に走査を終了する．s_i 番目の文番号が C に含まれている場合，これは該当 Wikipedia 記事の各文の TF-IDF ベクトルと会話データからなる TF-IDF ベクトルのコサイン類似度によってランサーが抽出した有益な記述の抽出を試みる場合，コサイン類似度が高い順に s_i 番目として有益な記述が出現することを表す．ただし，w_{s_i} に対応するコサイン類似度が 0 の場合は，TF-IDF ベクトルのコサイン類似度では有益な記述を取り出せなかったとみなす．

ここまでの操作をステップ 2 でランサーが有益だと判断した 681 件およびステップ 3 で過半数を超えて有益であると判断された 389 件に適用した結果が表 4.7 である．681 件について TF-IDF ベクトルのコサイン類似度により有益な記述の特定を試みると，平均 22.08 番目に，389 件について同様の操作を試みると，平均 20.62 番目にランサーが有益だと判断した記述が出現する．また，同方法によって第 1 位として有益な記述が抽出された件数は，681 件の記述については 59 件，389 件については 33 件であり，どちらについても 9%未満（抽出できた件数に占める割合では 13%未満）しか第 1 位として有益な記述が抽出できていないことがわかる．

以上より，本研究で構築したコーパスは TF-IDF のようなシンプルな方法では抽出できない記述を抽出したコーパスであると言える．

[1] http://taku910.github.io/mecab/

表 4.7: 有益な記述を TF-IDF ベクトルのコサイン類似度で抽出する操作の結果

	ステップ 2 で獲得した有益な記述 681 件	ステップ 3 で過半数が有益と判断した記述 389 件
コサイン類似度 0 の件数	217	110
有益な記述を抽出できた（コサイン類似度が 0 ではない）件数	464	279
有益な記述を抽出できた際に，W_{sort} で平均何番目に出現するか	22.08	20.62
W_{sort} で 1 番目に（第 1 位として）有益な記述が出現した数	59	33

4.5 考察

本研究では，最長マッチにより記事名を抽出した．4.1 節で述べたように，抽出される記事名の中には，ひらがな 1 文字の記事や記号など話題となりにくい語が多く含まれる．これらを取り除く手法を作成することで，上記のような記事をあらかじめ削除し，ステップ 1 からクラウドソーシングを用いたより客観的なコーパスを構築できると考えられる．

その手法としては，日本語 Wikification をベースとする手法が考えられる．アノテーション候補となる記事名を減らしつつ，最長マッチではリンクさせることができない記事名を抽出するということである．しかし，会話データに対する日本語 Wikifaciton およびそのためのコーパス構築に関する研究は現状存在しない．Murawaki ら [5] が構築した日本語 Wikification コーパスは，Twitter データおよび現代日本語書き言葉均衡コーパス（BCCWJ）の白書とブログを利用した日本語 Wikification コーパスであり，Jargalsaikhan ら [3] にもよるものは，BCCWJ の新聞記事を利用したものである．会話データに対する Wikification の発展により，例えば，本提案手法では取り除いた曖昧さ回避ページについても，適切な記事にリンクさせることで，話題として利用できる記事，有益な記述は増加すると考えられる．

本研究では，ステップ 2 およびステップ 3 での 2 段階のクラウドソーシングによってコーパスを構築した．クラウドソーシングについて，専門家によるアノテーションとクラウドソーシングを利用したアノテーションを比較して，クラウドソーシングの性能を評価した研究がなされている [22]，[23]．

Snow ら [22] は，自然言語処理の 5 種類のタスクにクラウドソーシングを適用し，クラウドソーシングによるアノテーションと専門家によるアノテーションとを比較を行っている．例えば，テキスト中の感情を推定し，6 つの感情についてスコア付けするタスクでは，クラウドソーシングによるアノテーターを 4 人集めれば，専門家 1 人に匹敵するアノテーションが行えるとしている．

自然言語処理とは異なるが Nowak ら [23] は，画像に対するアノテーションについて，画像データセットの詳細を与えられたアノテーター 11 人と，クラウドソーシングによる，データセットの詳細を与えられないアノテーターとによる，アノテーション結果の違いについて言及している．各アノテーションについて，データセット詳細を与えられた群とクラウドソーシングによる群とで，それぞれ多数派を ground-truth とした際の一致率が 0.92 であったと報告している．

本研究では，ステップ 2 で提出された明らかに有益ではない記述をステップ 3 に参加したランサーが有益ではないと判断するなど，2 段階に分けたクラウドソーシングの有用性を確認しているが，一方で，ステップ 2 には明らかに有益ではない記述が有益であるとして提出されたタスク遂行結果が，ステップ 3 にはスパムによるタスク遂行結果が含まれていた．

Kittur ら [24] は，Wikipedia の 1 つの記事を読ませ 6 つの項目について評価を行わせるタスクをクラウドソーシングによって行っている．Kittur らは，平均して 1 分 30 秒でタスクが遂行されており，タスク遂行数 210 件のうち 64 件が 1 分以内にタスクを終えていたと報告しており，Wikipedia 記事を読み 6 項目への評価を行うには短い時間であると述べている．また，これを踏まえ，タスクに真面目に取り組むか否かでワーカーの負担が変わらないような設計が良いとも述べている．Wikipedia を読ませるタスクという点で，これは本研究のステップ 2 と関連する．

Lancers では，各タスクに費やされた時間を計測することはできないが，有益であるとしたランサーにのみ有益な記述を抽出する負担が発生し，有益ではないとした方が労力が少なく済むタスク設計であることから，ステップ 2 において Wikipedia がランサーに最後まで読まれていないことも十分に考えられる．4.2 節と 4.3 節を通した挙げたステップ 3 で有益と判断される記述が 5 人中 4 人に（4 人が手抜きをせず最後まで記事を読んだかを知る方法はないが）ステップ 2 で見逃されていることも，これを示唆している．

同じく Kittur ら [24] は画像や段落の数を数えるといった質問を用意することでタスクに取り組む時間が増加したとも報告している．ステップ 2 が有益な記述の抽出を目的とすることからも，Kittur らが指摘するようなタスク設計で，より多くの有益な記述が獲得できる可能性がある．

また，スパムを未然に防ぐことはもちろん重要であるが，クラウドソーシング終了後に，各アノテーターが持つ能力を考慮することで，アノテーション結果に補正をかける研究が存在する [25]，[26]．本提案手法のステップ 2 やステップ 3 では，多数決により有益であるか有益ではないかの判定を行うことはできるが，スパムや誤ったタスク遂行結果がしばしば含まれるため，特にステップ 2 では多数決で該当する Wikipedia 記事が会話する 2 者にとって有益であるか判別することはできないと 4.2 節で述べた．

このような単純な多数決でラベルを決定できない事態への対策として，Whitehill ら [25] はアノテーターの能力やタスクごとの難しさを考慮した EM アルゴリズムによって，最終的なラベルを決定する方法を提案している．Ipeirotis ら [26] は，バイアスがかかった上でアノテーションを行う人を考慮に入れ，ラベルを決定する方法を提案している．

4.3 節では，有益だと判断する基準が異なることや，有益であるか有益ではないかの票が別れる記述が多いことを確認した．そのため，例えば，対話システムのユーザが持つ事前知識やシステムとの会話履歴などを利用し，単純な多数決に依らずにユーザにとって，有益な記述であるか有益ではないかを決定することが，本研究で構築したコーパスの利用した機械

学習手法を検討する際に重要となると考えられる.

第5章　結論

本研究では，Wikipedia を用いることで会話文中の話題を同定するためのコーパスを 3 つのステップによって構築した．このコーパスでは，会話中の発話を，話題を表す記事として “会話している人らにとって有益な記述” が存在する Wikipedia 記事に結びつけている．

コーパス構築の 1 ステップ目は，1,000 件の会話データに含まれる Wikipedia 記事の見出し文を確認し，アノテーター 1 人が有益な記述が含まれうる記事か判断するステップであり，本研究では著者がこのステップに取り組んだ．2 ステップ目は，1 ステップ目で有益な記述が含まれうると判断された記事内に，実際に有益な記述が含まれるかをクラウドソーシングにより判断するステップであり，有益と判断する場合には有益な記述の抽出も行わせる．3 ステップ目は，2 ステップ目で抽出された有益な記述についてどれだけの合意が得られるか，再度クラウドソーシングを利用して確認するステップである．この 3 ステップによって構築したコーパスは，ステップ 1 でのアノテーション結果を表す 1,000 件のテキストデータ，ステップ 2 で獲得した有益な記述を収録した json データ，ステップ 3 で獲得した，有益な記述についてどれだけ合意が得られるかを収録した json データによって構成される．

コーパス構築後は，構築したコーパスについて分析を行い，同じ記述でも有益であるか有益ではないかの判断が人によって分かれることがあるなど，人によって有益とする判断基準が異なることを確認した．加えて，同じ発話から抽出された同じ記事内の記述であっても，有益であるとするランサーの数が大きく異なる場合があることを確認した．また，クラウドソーシングの 1 段階目（ステップ 2）で抽出された記述が明らかに誤っている場合に，2 段階目（ステップ 3）で正しく有益ではないと判断されるなど，クラウドソーシングを 2 段階に分けることの有用性を確認した．

本研究で構築したコーパスは，発話中の話題を同定する機械学習，会話の話題の変化を追う機械学習，挨拶のみの会話など Wikipedia を参照する必要がない会話の判別を行う機械学習や Wikipedia 記事中の有益な記述を判別する機械学習を検討していく際に利用できると考えられる．本研究で構築したコーパスが，これらの機械学習手法の発展に貢献し，非タスク指向型対話システムがより多くの話題を扱う上での足がかりになることが期待される．

謝辞

若林啓先生．博士前期課程に 3 年属する私を最後まで見限ることなく丁寧にご指導くださったこと誠に感謝しております．大事なのは自分がどうしたいかですよ，と言葉をいただいたことは忘れません．学類時代から数えると 4 年間お世話になりました．誠にありがとうございました．

手塚太郎先生．いつも明るく，されどゼミや合宿で私が発表をした際は真剣な眼差しで発表を聞いてはアドバイスをくださり，大変励みとなりました．誠にありがとうございました．

slis16 の皆様，特に，野沢建人さん，福田拓也さん．立ち止まってばかりの私ではありましたが，つくばの地を離れても Slack 上で励ましのメッセージを送ってくださり，大変力が湧きました．誠にありがとうございました．

また手塚若林研究室の M2，平松淳さん，福山怜史さん，菊池祥平さん，宮原捺希さん．3 年間かかってしまった私ですが暖かく接してくださり，ここまで修論に取り組むことができました．誠にありがとうございました．

春日エリアキャリア相談室キャリアカウンセラー神村孝子先生．度々進路に悩む私にアドバイスをくださり，誠にありがとうございました．学類 4 年次からお世話になり，おかげさまで，納得する進路を選択することができました．

皆様，本当にありがとうございました．

参考文献

[1] 小磯花絵, 土屋智行, 渡部涼子, 横森大輔, 相澤正夫, 伝康晴. 均衡会話コーパス設計のための一日の会話行動に関する基礎調査. 国立国語研究所論集, No. 10, pp. 85–106, 2016.

[2] Rada Mihalcea and Andras Csomai. Wikify! Linking Documents to Encyclopedic Knowledge. In *Proceedings of the 16th Association for Computing Machinery Conference on Information and Knowledge Management (CIKM'07)*, pp. 233–242, 2007.

[3] Davaajav Jargalsaikhan, 岡崎直観, 松田耕史, 乾健太郎. 日本語 Wikification コーパスの構築に向けて. 言語処理学会 第 22 回年次大会発表論文集, pp. 793–796, 2016.

[4] 松田耕史, 岡崎直観, 乾健太郎. 日本語 wikification ツールキット: jawikify. 言語処理学会 第 23 回年次大会発表論文集, pp. 250–253, 2017.

[5] Yugo Murawaki and Shinsuke Mori. Wikification for Scriptio Continua. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, pp. 1346–1351, 2016.

[6] Seokhwan Kim, Rafael E Banchs, and Haizhou Li. Wikification of Concept Mentions within Spoken Dialogues Using Domain Constraints from Wikipedia. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*, pp. 2225–2229, 2015.

[7] Ahmet Yıldırım, Suzan Üsküdarlı, and Arzucan Özgür. Identifying Topics in Microblogs Using Wikipedia. *PLOS ONE*, Vol. 11, No. 3, pp. e0151885_1–20, 2016.

[8] Yegin Genc, Yasuaki Sakamoto, and Jeffrey V. Nickerson. Discovering Context: Classifying Tweets through a Semantic Transform Based on Wikipedia. In *Proceedings of the 6th International Conference on Foundations of Augmented Cognition (FAC'11)*, pp. 484–492, 2011.

[9] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the 19th Association for Computing Machinery Conference on Information and Knowledge Management (CIKM'10)*, pp. 1625–1628, 2010.

[10] Paolo Ferragina and Ugo Scaiella. Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *Institute of Electrical and Electronics Engineers Software (IEEE Software)*, Vol. 29, No. 1, pp. 70–75, 2012.

[11] Koichiro Yoshino and Tatsuya Kawahara. Conversational system for information navigation based on POMDP with user focus tracking. *Computer Speech and Language*, Vol. 34, No. 1, pp. 275–291, 2015.

[12] 石田真也, 井上昂治, 中村静, 高梨克也, 河原達也ほか. 傾聴対話システムのための発話を促す聞き手応答の生成. 人工知能学会 言語・音声理解と対話処理研究会（SIG-SLUD）第 77 回研究会, pp. 1–6, 2016.

[13] 功刀雅士, 若林啓. コンテキストを考慮した非タスク指向型対話システムの構築. 2016 年度人工知能学会全国大会 (JSAI2016) 論文集, pp. 1J3-4in1_1–4, 2016.

[14] Graham Wilcock. Wikitalk: A Spoken Wikipedia-based Open-Domain Knowledge Access System. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING'12)*, pp. 57–69, 2012.

[15] 別所克人, 東中竜一郎, 大塚淳史, 牧野俊朗, 松尾義博. 雑談対話における話題継続願望判定の検討. 人工知能学会 言語・音声理解と対話処理研究会（SIG-SLUD）第 74 回研究会, pp. 1–6, 2015.

[16] 東中竜一郎, 船越孝太郎, 荒木雅弘, 塚原裕史, 小林優佳, 水上雅博. Project Next NLP 対話タスク: 雑談対話データの収集と対話破綻アノテーションおよびその類型化. 言語処理学会 第 21 回年次大会ワークショップ論文集, pp. 1–12, 2015.

[17] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of Wikipedia: Knowledge-Powered Conversational Agents. In *7th International Conference on Learning Representations (ICLR'19)*, pp. 1–18, 2019.

[18] Martin Potthast. Crowdsourcing a Wikipedia Vvandalism Corpus. In *Proceedings of the 33rd international Association for Computing Machinery Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval (SIGIR'10)*, pp. 789–790, 2010.

[19] Elena Filatova. Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, pp. 392–398, 2012.

[20] 塚原裕史, 内海慶. オープンプラットフォームとクラウドソーシングを活用した対話コーパス構築方法. 言語処理学会 第 21 回年次大会発表論文集, pp. 147–150, 2015.

[21] 河原大輔, 町田雄一郎, 柴田知秀, 黒橋禎夫, 小林隼人, 颯々野学ほか. 2 段階のクラウドソーシングによる談話関係タグ付きコーパスの構築. 情報処理学会第 217 回自然言語処理研究発表会, pp. 1–7, 2014.

[22] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and Fast—But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, pp. 254–263, 2008.

[23] Stefanie Nowak and Stefan Rüger. How Reliable are Annotations via Crowdsourcing A Study about Inter-annotator Agreement for Multi-label Image Annotation. In *Proceedings of the International Conference on Multimedia Information Retrieval (MIR'10)*, pp. 557–566, 2010.

[24] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing User Studies With Mechanical Turk. In *Proceedings of the Special Interest Group on Computer-Human Interaction Conference on Human Factors in Computing Systems (CHI'08)*, pp. 453–456, 2008.

[25] Jacob Whitehill, Ting fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Advances in Neural Information Processing Systems 22*, pp. 2035–2043. 2009.

[26] Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang. Quality Management on Amazon Mechanical Turk. In *Proceedings of the Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining Workshop on Human Computation (HCOMP'10)*, pp. 64–67, 2010.

付 録A　Wikipediaダンプデータについて

2017 年 6 月 20 日の日本語版 Wikipedia のダンプデータ [1] における記事本文の xml は 155,969,130 行でありそのまま扱うには困難である．そのため，本研究では xml を記事ごとに分割し，本文を抽出した DB を作成し，利用している．ただし，後述する特殊記事とリダイレクト記事はこの DB には含まない．ダンプデータの例として，ソースコード A.1 に，2017 年 6 月 20 日の日本語版 Wikipedia のダンプデータにおける記事 “ドジっ娘” の xml を記す．ただし，各行の先頭には説明のため行番号を付与してある．

記事名

記事名は`<title>`タグで表現されている（ソースコード A.1 では 2 行目）．Wikipedia 記事には，非タスク指向型対話に利用するには不向きな記事が存在し，本研究ではこれを “特殊記事” と定義し DB に含めないことにしている．具体的には記事名が “`Category:`”，“`Wikipedia:`”，“`Template:`”，“`Portal:`”，“ファイル:”，
“`MediaWiki:`”，“`Help:`”，“モジュール:”，“プロジェクト:” から始まる記事を本研究では特殊記事としている．特殊記事に含まれるものには，例えば，日本語版 Wikipedia の記事数などの統計をまとめた記事 “`Wikipedia:`日本語版の統計” や Wikipedia の記事編集方法について述べる記事 “`Help:`ページの編集” がある．

ダンプデータにおける本文

`<text>`タグに囲まれた箇所（ソースコード A.1 では 15 行目から 41 行目）は “ダンプデータにおける本文” である．本文は構造化されておらず，例えば，本文中の “`==` 概要 `==`” や “`==` 脚注 `==`” のように等号で囲まれた文字列は，“セクション” を表しており，等号の数によりセクションの深さが表現される．また，`<text>`タグ中の最初のセクション（この例では “`==` 概要 `==`”）までが見出し文を表す．

[1] https://dumps.wikimedia.org/jawiki/

ソースコード A.1: Wikipedia ダンプデータの例

```
<page xmlns="http://www.mediawiki.org/xml/export-0.10/">
    <title>ドジっ娘</title>
    <ns>0</ns>
    <id>496931</id>
    <revision>
      <id>56710363</id>
      <parentid>50664915</parentid>
      <timestamp>2015-09-02T06:15:48Z</timestamp>
      <contributor>
        <ip>2001:268:C036:5556:BDD9:20F7:BFDB:B886</ip>
      </contributor>
      <comment>/* 関連項目 */</comment>x
      <model>wikitext</model>
      <format>text/x-wiki</format>
      <text xml:space="preserve">[[File:Wikipe-tanCrazy.gif|thumb|right|220px|立体ジグソーパズルの組み立てに失敗する [[Wikipedia:ウィキペたん|ウィキペたん]]。]]
[[File:Dojikko.png|thumb|right|220px|ドジっ娘に物を運ばせるとこうなる。]]
''' ドジっ娘'''（ドジっこ）は、ドジな [[女性]] を総称して指す用語&lt;ref name="ラノベ超 150"&gt;『ライトノベル「超」入門(新城カズマ、ソフトバンク新書、ISBN 4797333383)』p.150&lt;/ref&gt; で、[[オタク]] カルチャー用語。

''' ドジっ子'''、''' ドジ女''' とも用いる。またドジっ子という言葉であると、[[男性]][[キャラクター]] にも当てはまる。

== 概要 ==
家事一般、スポーツ競技、通常歩行を含む運動全般などにおいて、可愛らしいあるいは憎めない失敗を繰り返す女性キャラクターを指す&lt;ref name="ラノベ超 150" /&gt;。

ドジっ娘が行う所作の一例として、『少女マンガから学ぶ恋愛学』では「階段から滑り落ちる」「コーヒーをひっくり返す」など&lt;ref&gt;『少女マンガから学ぶ恋愛学』(架神恭介、シンコーミュージック、ISBN 4401630904)p.67&lt;/ref&gt; が、『オタク用語の基礎知識』では「道端で転ぶ」「接客業で皿を割る」など&lt;ref&gt;『オタク用語の基礎知識』(オタク文化研究会、マガジンファイブ、ISBN 4434073966)p.87&lt;/ref&gt; が挙げられている。

[[アニメ]]・[[コンピュータゲーム|ゲーム]] をはじめとする [[サブカルチャー]] 系作品における [[萌え属性]] のひとつで、男性読者・視聴者の愛好対象となるよう設計されることが多いが、『[[エースをねらえ!]]』の岡ひろみや『[[美少女戦士セーラームーン]]』の [[月野うさぎ]] など、[[少女マンガ]] の主人公キャラクターにもこの属性が付与されることがしばしばある&lt;ref name="ラノベ超 150" /&gt;。

ドジっ娘に対し好意を抱き、惹かれるさまを「ドジっ娘萌え」と呼ぶ&lt;ref&gt;『世界はゴミ箱の中に(青木敬士、現代図書、ISBN 4434056786)』p.115&lt;/ref&gt;。

== 脚注 ==
{{Reflist}}

== 関連項目 ==
* [[ストックキャラクター]]
* [[天然ボケ]]
* [[ぶりっ子]]
* [[らいらん]]

{{DEFAULTSORT:としつこ}}
[[Category:萌え属性]]
[[Category:萌え文化におけるキャラクター類型]]</text>
      <sha1>5zo1ik4b8bht89rxfdqb3pvxykq7lq5</sha1>
    </revision>
  </page>
```

アンカー

本文中の“[[”と“]]”で囲まれた文字列はアンカーと呼ばれる．これは他の Wikipedia 記事へのリンクを表すもので，“[[アニメ]]”は記事“アニメ”へのリンクを表し，Web 上では“アニメ”と表示される．“[[コンピュータゲーム|ゲーム]]”のように“|”が含まれるアンカーでは，“|”の右側が Web 上で表示される文字列を，左側がリンク先の記事名を表している．

リダイレクト記事

リダイレクト記事は，一意に特定の記事へとリンクさせる記事である．ダンプデータでは<redirect title>タグでリダイレクト先の記事名が表現される．例えば，記事“Doraemon”は記事“ドラえもん”へのリダイレクト記事であり，Web 上で記事“Doraemon”へのアクセスを試みる場合，図 A.1 のように“（Doraemon から転送）”と表記され，記事“ドラ

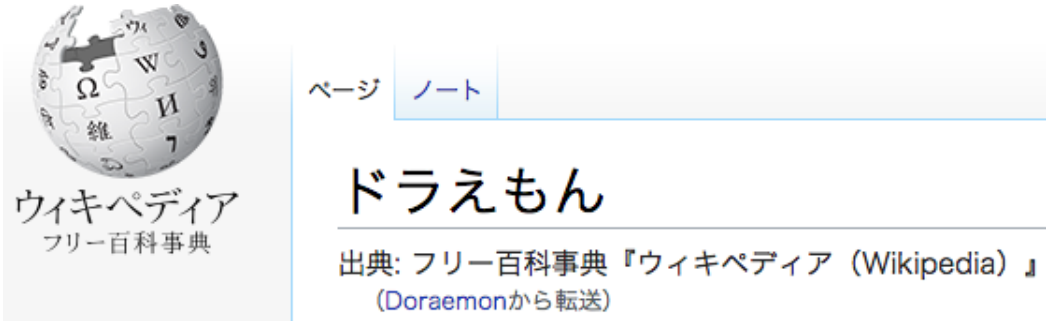

図 A.1: 記事 “Doraemon” から記事 “ドラえもん” へのリダイレクト (2018 年 12 月 3 日 14 時 41 分参照)

えもん” へと一意に転送される．“Doraemon” 以外にも記事 “ドラえもん” へのリダイレクト記事は存在し，2017 年 6 月 20 日の日本語版 Wikipedia のダンプデータにおいては，記事 “どらえもん”，記事 “ドラエモン”，記事 “ドラエもん”，記事 “Doraemon” が記事 “ドラえもん” へのリダイレクト記事である．一意に別の記事にリンクさせる性質から，3.2 節で利用する．なお，特殊記事へのリダイレクト記事を除いた場合，このダンプデータには 643,627 のリダイレクト記事が存在する．